Claims

1. Apparatus for configuration independent simulation of network layer conditions in a simulated network that transmits data packets between a DUT and another component, the apparatus comprising:

   a network layer verification mechanism connected between the DUT and the other component, the network layer verification mechanism having a storage and a plurality of methods for selectively forwarding data packets between the DUT and the other component or for selectively storing data packets moving between the DUT and the other component; and

   an API interface for invoking the methods to simulate conditions that can occur in the network, including dropped packets, duplicate packets, corrupted packets, out-of-order packets and delayed packets.

2. The apparatus of claim 1 wherein the network layer verification mechanism is implemented as a specialized object written in an HVL.

3. The apparatus of claim 2 wherein the object includes internal storage in the form of an associative array and a plurality of methods that allow packets received by the object to be selectively forwarded through the object, temporarily stored in the object or the packet data to be corrupted.

4. The apparatus of claim 1 wherein the network layer verification mechanism comprises a packet ingress section and a packet egress section.

5. The apparatus of claim 4 wherein the API interface includes a method for transmitting packets between the packet ingress section and the packet egress section.

6. The apparatus of claim 4 wherein the API interface comprises a method for transmitting packets between the packet ingress section and the storage.

7. The apparatus of claim 4 wherein the API interface comprises a method for transmitting a packet stored in the storage to the packet egress section.

8. The apparatus of claim 4 wherein the API interface comprises a method for retrieving a packet stored in the storage.

9. The apparatus of claim 4 wherein the API interface comprises a method for modifying a data packet received at the ingress section.

10. The apparatus of claim 9 wherein the API interface comprises a method for restoring a modified data packet in the storage.

11. A method for configuration independent simulation of network layer conditions in a simulated network that transmits data packets between a DUT and another component, the method comprising:

(a) creating a network layer verification mechanism having a storage and a plurality of methods for selectively forwarding data packets between the DUT and the other component or for selectively storing data packets moving between the DUT and the other component;

(b) connecting the network layer verification mechanism between the DUT and the other component; and

(c) invoking the methods with an API interface to simulate conditions that can occur in the network, including dropped packets, duplicate packets, corrupted packets, out-of-order packets and delayed packets.

12. The method of claim 11 wherein step (a) comprises implementing the network layer verification mechanism as a specialized object written in an HVL.

13. The method of claim 12 wherein step (a) comprises creating the object with internal storage in the form of an associative array and a plurality of methods that allow packets received by the object to be selectively forwarded through the object, temporarily stored in the object or the packet data to be corrupted.

14. The method of claim 11 wherein step (a) comprises receiving packets at a packet ingress section of the network layer verification mechanism and transmitting packets from a packet egress section of the network layer verification mechanism.

15. The method of claim 14 wherein the API interface includes a method for transmitting packets between the packet ingress section and the packet egress section.

16. The method of claim 14 wherein the API interface comprises a method for transmitting packets between the packet ingress section and the storage.

17. The method of claim 14 wherein the API interface comprises a method for transmitting a packet stored in the storage to the packet egress section.

18. The method of claim 14 wherein the API interface comprises a method for retrieving a packet stored in the storage.

19. The method of claim 14 wherein the API interface comprises a method for modifying a data packet received at the ingress section.

20. The method of claim 19 wherein the API interface comprises a method for restoring a modified data packet in the storage.

21. A computer program product for configuration independent simulation of network layer conditions in a simulated network that transmits data packets between a DUT and another component, the computer program product comprising a computer usable medium having computer readable program code thereon, including:

(a) program code that creates a network layer verification mechanism having a storage and a plurality of methods for selectively forwarding data packets between the DUT and the other component or for selectively storing data packets moving between the DUT and the other component;

(b) program code that connects the network layer verification mechanism between the DUT and the other component; and

(c) an API interface that invokes the methods to simulate conditions that can occur in the network, including dropped packets, duplicate packets, corrupted packets, out-of-order packets and delayed packets.

22. The computer program product of claim 21 wherein the program code that creates a network layer verification mechanism comprises program code that implements the network layer verification mechanism by instantiating a as a specialized object written in an HVL.

23. The computer program product of claim 22 wherein the program code that creates a network layer verification mechanism comprises program code that instantiates the object with internal storage in the form of an associative array and a plurality of methods that allow packets received by the object to be selectively forwarded through the object, temporarily stored in the object or the packet data to be corrupted.

24. The computer program product of claim 21 wherein the program code that creates a network layer verification mechanism comprises program code that creates a packet ingress section of the network layer verification mechanism and

program code that creates a packet egress section of the network layer verification mechanism.

25. The computer program product of claim 24 wherein the API interface includes a method for transmitting packets between the packet ingress section and the packet egress section.

26. The computer program product of claim 24 wherein the API interface comprises a method for transmitting packets between the packet ingress section and the storage.

27. The computer program product of claim 24 wherein the API interface comprises a method for transmitting a packet stored in the storage to the packet egress section.

28. The computer program product of claim 24 wherein the API interface comprises a method for retrieving a packet stored in the storage.

29. The computer program product of claim 24 wherein the API interface comprises a method for modifying a data packet received at the ingress section.

30. The computer program product of claim 29 wherein the API interface comprises a method for restoring a modified data packet in the storage.

31. A computer data signal embodied in a carrier wave for configuration independent simulation of network layer conditions in a simulated network that transmits data packets between a DUT and another component, the computer data signal comprising:

(a) program code that creates a network layer verification mechanism having a storage and a plurality of methods for selectively forwarding data packets between the DUT and the other component or for selectively storing data packets moving between the DUT and the other component;

(b) program code that connects the network layer verification mechanism between the DUT and the other component; and

(c) an API interface that invokes the methods to simulate conditions that can occur in the network, including dropped packets, duplicate packets, corrupted packets, out-of-order packets and delayed packets.